

**MEMORY HUB ARCHITECTURE HAVING PROGRAMMABLE LANE
WIDTHS**

TECHNICAL FIELD

[001] This invention relates to processor-based systems having system memory, and, more particularly, to processor-based systems having memory modules containing a memory hub in which the memory hubs are coupled to the processor through a memory hub controller through a downstream bus and an upstream bus.

BACKGROUND OF THE INVENTION

[002] Processor-based systems, such as computer systems, use memory devices, such as dynamic random access memory (“DRAM”) devices, as system memory to store instructions and data that are accessed by a processor. In a typical computer system, the processor communicates with the system memory through a processor bus and a memory controller. The processor issues a memory request, which includes a memory command, such as a read command, and an address designating the location from which data or instructions are to be read or to which data or instructions are to be written. The memory controller uses the command and address to generate appropriate command signals as well as row and column addresses, which are applied to the system memory. In response to the commands and addresses, data is transferred between the system memory and the processor. The memory controller is often part of a system controller, which also includes bus bridge circuitry for coupling the processor bus to an expansion bus, such as a PCI bus.

[003] Although the operating speed of memory devices has continuously increased, this increase in operating speed has not kept pace with increases in the operating speed of processors. Even slower has been the increase speed at which memory commands, addresses and data can be coupled between processors and

memory devices. The relatively slow speed of memory controllers and memory devices limits the data bandwidth between the processor and the memory devices.

[004] One approach to increasing the data bandwidth to and from memory devices is to use multiple memory devices coupled to the processor through a memory hub as shown in Figure 1. A computer system 100 using a memory hub architecture includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 includes a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is typically coupled to cache memory 108, which, is typically static random access memory (“SRAM”). Finally, the processor bus 106 is coupled to a system controller 110, which is also sometimes referred to as a bus bridge.

[005] The system controller 110 contains a memory hub controller 128 that is coupled to the processor 104. The memory hub controller 128 is also coupled to several memory modules 130a-n through a bus system 134. Each of the memory modules 130a-n includes a memory hub 140 coupled to several memory devices 148 through command, address and data buses, collectively shown as bus 150. The memory hub 140 efficiently routes memory requests and responses between the controller 128 and the memory devices 148. Computer systems employing this architecture can have a higher bandwidth because the processor 104 can access one memory module 130a-n while another memory module 130a-n is responding to a prior memory access. For example, the processor 104 can output write data to one of the memory modules 130a-n in the system while another memory module 130a-n in the system is preparing to provide read data to the processor 104. The operating efficiency of computer systems using a memory hub architecture can make it more practical to vastly increase data bandwidth of a memory system. A memory hub architecture can also provide greatly increased memory capacity in computer systems.

[006] The system controller 110 also serves as a communications path to the processor 104 for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller

112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or more input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

[007] A memory hub architecture can greatly increase the rate at which data can be stored in and retrieved from memory because the point-to-point nature of the bus system 134 allows faster electrical signaling, and because the isolated nature of the DRAMs allows parallel or overlapping activity in several modules. In fact, a memory system using several memory modules each containing a memory hub can collectively transmit and receive data at such a high rate that the bus system 134 can become the “bottleneck” limiting the data bandwidth of the memory system.

[008] One technique that has been used to maximize the data bandwidth of memory systems using a memory hub architecture is to implement the bus system 134 using a high-speed “downstream” bus 154 and an “upstream” bus 156 that is separate from the downstream bus 154. The downstream bus 154 and the upstream bus have the same width, *i.e.*, number of conductors. The high-speed downstream bus 154 couples packets containing a memory command, an address, and write data from the memory hub controller 128 to the memory modules 130 and from the memory modules 130 to memory modules 130 located further away from the memory hub controller 128. The high-speed upstream bus 156 couples a packet containing read data and an identifier from memory modules 130 to the memory hub controller 128 and from the memory modules 130 to memory modules 130 located closer to the memory hub controller 128. The use of two different uni-directional buses has the advantage of eliminating turn

around latency and allowing the memory hub controller to simultaneously transmit and receive data.

[009] The data bandwidth of memory systems using a memory hub architecture can also be maximized by using a different type of memory signaling protocol. Rather than using traditional address, data and control buses, the address, data and control bits for each memory request or “transaction” are sent together in a single packet. The packet includes a command header followed by read or write data. The command header includes bits corresponding to a memory command, such as a write or a read command, identifying bits that specify the memory module to which the request is directed, and address bits that specify the address of the memory devices 148 in the specified memory module that is being accessed with the request. The command header may also specify the quantity of read or write data that follows the command header. The use of a packetized memory system allows the memory hub controller 128 to issue a memory request by simply transmitting a packet instead of transmitting a sequence of command, address and, in the case of a write request, write data signals. As a result, the memory hub controller 128 can issue memory requests at a faster rate. Furthermore, a packetized memory system frees the memory hub controller 128 from having to keep track of the processing details of each memory request. Instead, the memory hub controller 128 need only transmit the packet. The memory hub 140 in the memory module 130 to which the memory request is directed then processes the memory request without further interaction with the memory hub controller 128. In the case of a read request, the memory hub 140 transmits a packet back to the memory hub controller 128, either directly or through intervening memory modules 130, that contains the read data as well as identifying bits in a command header identifying the read data. The memory hub controller 128 uses the identifying bits to associate the read data with a specific memory request.

[010] Although the memory hub architecture shown in Figure 1 can provide a significant increase in the data bandwidth between the processor 104 and the memory devices 148, it nevertheless can provide at less than optimum performance. In

particular, the capacity of either the downstream bus 154 or the upstream bus 156 can limit the rate at which packets can be coupled to or from the memory modules 130, thereby limiting data bandwidth. The particular bus 154, 156 that becomes overloaded normally depends on the nature of the memory accesses. Memory writes result in a significantly greater amount of traffic through the downstream bus 154 compared to the traffic through the upstream bus 156. Memory reads still require that packets containing commands and addresses be coupled through the downstream bus 154, but they generally result in significantly more traffic being routed through the upstream bus 156. Therefore, memory accesses primarily consisting of writes tend to overload the downstream bus 154, and memory accesses consisting primarily of reads tend to overload the upstream bus 156. In either case, the result is a reduced data bandwidth between the processor 104 and the memory devices 148.

[011] There is therefore a need for a memory hub architecture that is more able to avoid overloading either the downstream bus or the upstream bus coupled between a memory hub controller and one or more memory hubs.

SUMMARY OF THE INVENTION

[012] A memory system that may be used in a processor-based system includes a memory hub controller coupled to at least one memory module having a memory hub and a plurality of memory devices coupled to the memory hub. Command, address and data signals are coupled from the memory hub controller to the memory hub using a communications path having a first capacity, and data signals are coupled from the memory hub to the memory hub controller using a communications path having a second capacity. Although the sum of the first capacity and the second capacity is a fixed value, the individual values of the first and second capacities are altered during the operation of the memory system, such as while the memory system is being initialized, depending upon the magnitudes of the actual or anticipated signal flow through the communication path. In the event of an increase in the magnitude of the actual or anticipated signal flow from the memory hub controller to the memory hub,

the first capacity can be increased and the second capacity can be decreased. In the event of an increase in the magnitude of the actual or anticipated signal flow from the memory hub to the memory hub controller, the second capacity can be increased and the first capacity can be decreased.

BRIEF DESCRIPTION OF THE DRAWINGS

[013] Figure 1 is a block diagram of a processor-based system having system memory that uses a conventional memory hub architecture.

[014] Figure 2 is a block diagram of a processor-based system having system memory using a memory hub architecture according to one embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[015] A processor-based system 200 according to one embodiment of the invention is shown in Figure 2. The system 200 uses most of the same components used in the system 100 of Figure 1, and they operate in the same manner. Therefore, in the interest of brevity, they have been provided with the same reference number, and an explanation of their operation will not be repeated. The processor-based system 200 differs from the system 100 shown in Figure 1 in that the widths of the downstream bus 154 and upstream bus 156 are not fixed, nor are they necessarily equal to each other. Instead, the downstream bus 154 has a width of M bits, and the upstream bus 156 has a width of N bits, where the sum of M and N is equal to a fixed value. The values of M and N are dynamic, preferably depending on the nature of the actual or expected memory accesses. If a disproportionately large number of writes are occurring or expected to occur, the value of M is increased to increase the capacity of the downstream bus 154, and the value of N is decreased accordingly to decrease the capacity of the upstream bus 156. If a disproportionately large number of read are occurring or expected to occur, the value of M is decreased to decrease the capacity of

the downstream bus 154, and the value of N is increased accordingly to increase the capacity of the upstream bus 156.

[016] The capacity M of the downstream bus 154 relative to the capacity N of the upstream bus 156 can be determined using a variety of techniques. The system 200 can be manually configured in a semi-permanent manner by manually manipulating hardware jumpers or the like to select the values of M and N. In such case, the values of M and N may be determined during system design based on the anticipated usage of the system 200. However, rather than being static, the values of M and N are preferably dynamic, *i.e.*, they change during the operation of the system 200. The values of M and N may be dynamically changed based on either the actual traffic flow through the buses 154, 156, or the anticipated traffic flow through the buses 154, 156. Alternatively, the values of M and N may be changed at other times during the operation of the memory system, such as when the memory system is being initialized.

[017] If the actual traffic flow is used as the basis for adjusting M and N, the traffic through the buses 154, 156 may be determined by the memory hub controller 128 and the memory hubs 140, which can then adjust M and N by configuring internal buffers coupled to the signals lines in the buses 154, 156 as either input buffers or output buffers. Alternatively, the traffic through the buses 154, 156 may be determined by only the memory hub controller 128, which can couple configuration data to the memory hubs 140. In such case, default values for M and N are used to couple the configuration to the memory hubs 140. The memory hubs 140 can then use the configuration data to configure internal buffers coupled to the signals lines in the buses 154, 156 as either input buffers or output buffers. As an alternative to using hardware to analyze the traffic flow through the buses 154, 156, the actual traffic flow could be determined by software, such as the operation system or run time agents. Other means of determining the actual traffic flow through the buses 154, 156 can also be used.

[018] If anticipated traffic flow through the buses 154, 156 is used as the basis for adjusting M and N, the anticipated traffic flow may be determined by a user and then coupled to the memory hub controller 128 and the memory hubs 140 by suitable

means. For example, the anticipated usage information can be coupled through a suitable interface, such as a JTAG interface (not shown) or and I2C interface (not shown). In either case, the memory hub controller 128 and the memory hubs 140 can then configure internal buffers coupled to the signals lines in the buses 154, 156 accordingly.

[019] The anticipated traffic flow through the buses 154, 156 may be either determined directly by a user, as explained above, or it may be inferred from other information. The anticipated traffic flow the buses 154, 156 may be determined based on either the type of hardware included in the system 200 or the software being executed by the processor 104. For example, the processor 104 may determine the appropriate values of M and N and couple these values to the memory hub controller 128 and the memory hubs 140 based on the type of application program executed by the processor 104. There are normally a large number of write accesses when an installation program is being executed. In such case, the processor 104 would increase the value of M and decrease the value of N. A graphics intensive application program, on the other hand, normally requires a disproportionate number of read accesses, so the value of M would be decreased and the value of N would be increased. A communications program normally has about an equal number of reads and writes, so the value of M would be set equal to the value of N. Other hardware or software means for determining the expected traffic flow through the buses 154, 156 can alternatively be used.

[020] Although all of the signal lines in the buses 154, 156 may be configurable as bi-directional so that they can be either part of the downstream bus 154 or part of the upstream bus 156, some of the lines in the buses 154, 156 corresponding in number to minimum values of M and N may be uni-directional and thus dedicated to either the downstream bus 154 or the upstream bus 156. The requirement that the input of an input buffer be coupled to each line that is also coupled to the output of an output buffer may have a tendency in some cases to overload the line. However, this potential problem can be alleviated by equalization techniques or by decoupling the input buffers

from the lines of the buses 154, 156 when they are not in use, for example. The manner in which input and output buffers should be coupled to the buses 154, 156, as well as the specific hardware and software than can be used to determine the values of M and N, will be readily apparent to one skilled in the art.

[021] From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, it will be understood by one skilled in the art that various modifications may be made without deviating from the spirit and scope of the invention. For example, although the memory modules are shown as being physically separate from the memory hub controller, it will be understood that they need not be physically separate. Instead, the memory hub controller and the memory modules may be mounted on a common substrate, such as a motherboard. Accordingly, the invention is not limited except as by the appended claims.